




Operating System/2
Programming Tools
and Information
Version 1.2/1.3

C/2, COBOL/2, FORTRAN/2 and
Macro Assembler/2, Bindings Reference
Technical Upgrade

91F8683



C/2, COBOL/2, FORTRAN/2 and
Macro Assembler/2, Bindings Reference
Technical Upgrade

Operating System/2
Programming Tools
and Information
Version 1.2/1.3

First Edition (December 1990)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

COPYRIGHT LICENSE: This publication contains printed sample application programs in source language, which illustrate OS/2 programming techniques. You may copy and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the OS/2 application programming interface.

Each copy of any portion of these sample programs or any derivative work, which is distributed to others, must include a copyright notice as follows: "© (your company name) (year) All Rights Reserved."

© Copyright International Business Machines Corporation 1990. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

IBM
IBM C/2
IBM COBOL/2
IBM FORTRAN/2
IBM Macro Assembler/2
Operating System/2
OS/2
Presentation Manager
Systems Application Architecture.

About This Book

This book, as well as the other books in the upgrade package, contains programming information that is new to or different from the information in the *IBM® Operating System/2® (OS/2®) Programming Tools and Information Version 1.2, Building Programs* library (part number 64F0274). In addition, information from the *OS/2 Programming Tools and Information Version 1.2 Technical Update* (part number 64F1705) is included in this upgrade package.

The upgrade describes features added by the OS/2 Version 1.3 product, and amends some of the information that was published with OS/2 Version 1.2.

How To Use This Book

This book contains a section for the following Version 1.2 books:

Book Title	Book Number
Presentation Manager C/2 Bindings Reference	64F0278
Presentation Manager COBOL/2 Bindings Reference	64F0279
Presentation Manager FORTRAN/2 Bindings Reference	64F0280
Presentation Manager Macro Assembler/2 Bindings Reference	64F0281

Only those chapters of the Version 1.2 books that are changed by the Version 1.3 product are included herein. The chapter numbers and titles are the same as those in the Version 1.2 book. As a convenience, the beginning of each chapter in this book provides a summary of changes for that chapter.

Who Should Read This Book

This book is for application designers and programmers who are using the components of the *IBM OS/2 Version 1.2/1.3 Programming Tools and Information Technical Upgrade* package. The reader is assumed to be familiar with the services of OS/2.

* Trademark of the IBM Corporation

Contents

Presentation Manager C/2 Bindings Reference	1
Chapter 2. Data Types	3
DRAGIMAGE	3
DRAGINFO	4
DRAGITEM	4
DRAGTRANSFER	5
ESCSETMODE	5
HFILE	5
HLIB	5
HSTR	6
MARGSTRUCT	6
C language binding	6
MLE_SEARCHDATA	6
OVERFLOW	6
PAPSZ	6
PRDINFO	7
PRDINFO3	7
PRIDINFO	7
PRJINFO	8
PRJINFO2	8
PRJINFO3	9
PRQINFO	9
PRQINFO3	10
RENDERFILE	10
SPLERR	10
Chapter 3.1. Drag Function Calls	11
DrgAcceptDroppedFiles	12
DrgAccessDraginfo	12
DrgAddStrHandle	12
DrgAllocDraginfo	12
DrgAllocDragtransfer	12
DrgDeleteDraginfoStrHandles	13
DrgDeleteStrHandle	13
DrgDrag	13
DrgDragFiles	13
DrgFreeDraginfo	14
DrgFreeDragtransfer	14
DrgGetPS	14
DrgPostTransferMsg	14
DrgPushDraginfo	14
DrgQueryDragitem	15
DrgQueryDragitemCount	15
DrgQueryDragitemPtr	15
DrgQueryNativeRMF	15
DrgQueryNativeRMFLen	16
DrgQueryStrName	16
DrgQueryStrNameLen	16
DrgQueryTrueType	16

DrgQueryTrueTypeLen	16
DrgReleasePS	17
DrgSendTransferMsg	17
DrgSetDragImage	17
DrgSetDragItem	17
DrgSetDragPointer	17
DrgVerifyNativeRMF	18
DrgVerifyRMF	18
DrgVerifyTrueType	18
DrgVerifyType	18
DrgVerifyTypeSet	19
Chapter 7. Spooler Function Calls	21
DosPrintDestAdd	21
DosPrintDestControl	22
DosPrintDestDel	22
DosPrintDestEnum	22
DosPrintDestGetInfo	22
DosPrintDestSetInfo	23
DosPrintJobContinue	23
DosPrintJobDel	23
DosPrintJobEnum	23
DosPrintJobGetId	24
DosPrintJobGetInfo	24
DosPrintJobPause	24
DosPrintJobSetInfo	24
DosPrintQAdd	25
DosPrintQContinue	25
DosPrintQDel	25
DosPrintQEnum	25
DosPrintQGetInfo	26
DosPrintQPause	26
DosPrintQPurge	26
DosPrintQSetInfo	26
Chapter 10. Functions Supplied by Applications	27
JournalPlaybackHook	27

Presentation Manager COBOL/2 Bindings Reference 29

Chapter 1. Introduction	31
Named Constants	31
Chapter 2. Data Types	33
COUNT4	33
COUNT4B	33
LENGTH4	33
LONG	33
MARGSTRUCT	34
OVERFLOW	34
PROPERTY4	34
ULONG	34
WIDTH4	34
Chapter 8. Window Function Calls	35

WinQuerySwitchList — WMQSLI	35
<hr/>	
Presentation Manager FORTRAN/2 Bindings Reference	37
Chapter 1. Introduction	39
Named Constants	39
Chapter 2. Data Types	41
HPROGARRAY	41
MARGSTRUCT	41
OVERFLOW	41
Chapter 6. Profile Function Calls	43
PrfQueryProgramHandle — PRQPHD	43
Chapter 8. Window Function Calls	45
WinQuerySwitchList — WMQSLI	45
<hr/>	
Presentation Manager Macro Assembler/2 Bindings Reference	47
Chapter 2. Data Types	49
MARGSTRUCT	49
OVERFLOW	49
Chapter 9. Window Function Calls	51
WinQuerySessionTitle	51
WinSwitchToProgram	51
Chapter 10. Functions Supplied by Applications	53
JournalPlaybackHook	53
<hr/>	
Master Index	55
Index	57

Presentation Manager C/2 Bindings Reference

Chapter 2. Data Types

Summary of Changes

New	Updated	Section Title
✓		DRAGIMAGE
✓		DRAGINFO
✓		DRAGITEM
✓		DRAGTRANSFER
✓		ESCSETMODE
✓		HFILE
	✓	HLIB
✓		HSTR
	✓	MARGSTRUCT
	✓	MLE_SEARCHDATA
	✓	OVERFLOW
✓		PAPSZ
✓		PRDINFO
✓		PRDINFO3
✓		PRIDINFO
✓		PRJINFO
✓		PRJINFO2
✓		PRJINFO3
✓		PRQINFO
✓		PRQINFO3
✓		RENDERFILE
✓		SPLERR

DRAGIMAGE

Dragged-object-image structure.

```
typedef struct _DRAGIMAGE {  
    USHORT    cb;           /* Size, in bytes, of the DRAGIMAGE structure */  
    USHORT    cptl;         /* The number of points in the point array if fl  
                           is specified as DRG_POLYGON. */  
    LHANDLE    hImage;      /* Handle representing the image to display */  
    SIZE      sizlStretch;  /* Dimensions for stretching */  
    ULONG      fl;          /* Flags */  
    SHORT      cxOffset;    /* x offset from the pointer hot spot to the  
                           origin of the image */  
    SHORT      cyOffset;    /* y offset from the pointer hot spot to the  
                           origin of the image */  
}
```

```
} DRAGIMAGE;
```

DRAGINFO

Drag-information structure.

```
typedef struct _DRAGINFO {  
    ULONG      cbDraginfo;    /* Size, in bytes, of the structure */  
    USHORT     cbDragitem;    /* Size, in bytes, of each DRAGITEM structure */  
    SHORT      usOperation;    /* Modified drag operations */  
    HWND       hwndSource;    /* Window handle of the source of the drag  
                             operation */  
    SHORT      xDrop;         /* x coordinate of drop point expressed in  
                             desktop coordinates */  
    SHORT      yDrop;         /* y coordinate of drop point expressed in  
                             desktop coordinates */  
    USHORT     cditem;        /* Count of DRAGITEM structures */  
    USHORT     usReserved;    /* Reserved */  
}  
} DRAGINFO;
```

DRAGITEM

Drag-object structure.

```
typedef struct _DRAGITEM {  
    HWND       hwndItem;      /* Window handle of the source of the  
                             drag operation */  
    ULONG      ulItemID;      /* Information used by the source to  
                             identify the object being dragged */  
    HSTR       hstrType;       /* String handle of the object type */  
    HSTR       hstrRMF;        /* String handle of the rendering  
                             mechanism and format */  
    HSTR       hstrContainerName; /* String handle of the name of the  
                             container holding the source object */  
    HSTR       hstrSourceName; /* String handle of the name of the  
                             source object */  
    HSTR       hstrTargetName; /* String handle of the suggested name of  
                             the object at the target */  
    SHORT      cxOffset;      /* x offset from the pointer hot spot to  
                             the origin of the image that represents  
                             this object */  
    SHORT      cyOffset;      /* y offset from the pointer hot spot to  
                             the origin of the image that represents  
                             this object */  
    USHORT     fsControl;      /* Source-object control flags */  
    USHORT     fsSupportedOps; /* Direct manipulation operations  
                             supported by the source object */  
}  
} DRAGITEM;
```

DRAGTRANSFER

Drag-conversation structure.

```
typedef struct _DRAGTRANSFER {  
    ULONG        cb;                /* Size, in bytes, of the structure */  
    HWND         hwndClient;        /* Handle of the client window */  
    DRAGITEM     pditem;            /* The DRAGITEM structure that is to be  
                                   rendered */  
    HSTR         hstrSelectedRMF;    /* The string handle for the selected  
                                   rendering mechanism and format for the  
                                   transfer operation */  
    HSTR         hstrRenderToName;   /* A string handle representing the name  
                                   where the source will place, and the  
                                   target will find, the data that is  
                                   rendered */  
    ULONG        ulTargetInfo;       /* Reserved for use by the target */  
    USHORT       usOperation;        /* The operation */  
    USHORT       fsReply;            /* Reply flags for the message */  
} DRAGTRANSFER;
```

ESCSETMODE

Structure for setting printer mode.

```
typedef struct _ESCSETMODE {  
    ULONG        mode;              /* Mode */  
    USHORT       codepage;          /* Code page */  
} ESCSETMODE;
```

HFILE

Resource handle.

```
typedef USHORT HFILE;
```

HLIB

On page 2-10, change the definition to:

```
typedef HMODULE HLIB;
```

HSTR

String handle.

```
typedef &handle. HSTR;
```

MARGSTRUCT

C language binding

On page 2-12, this data type is now named *MLEMARGSTRUCT*.

Change the line that reads:

```
USHORT    fsFlags;
```

to:

```
USHORT    afFlags;
```

MLE_SEARCHDATA

On page 2-13, replace the description of this data type with:

```
typedef struct _MLE_SEARCHDATA {  
    USHORT    cb;           /* size of MLE_SEARCHDATA structure */  
    PCHAR      pchFind;     /* string to search for */  
    PCHAR      pchReplace;  /* string to replace with */  
    SHORT      cchFind;     /* length of pchFindString */  
    SHORT      cchReplace;  /* length of replace string */  
    IPT        iptStart;    /* point at which to start search */  
                                /* (negative indicates cursor pt) */  
                                /* becomes pt where string found */  
    IPT        iptStop;     /* point at which to stop search */  
                                /* (negative indicates EOT) */  
    USHORT     cchFound;    /* Length of found string at iptStart */  
} MLE_SEARCHDATA;
```

OVERFLOW

On page 2-14, this data type is now named *MLEOVERFLOW*.

PAPSZ

Pointer to an array of pointers to null-terminated strings.

```
typedef char far *PAPSZ;
```

PRDINFO

Print-destination information structure (level 1).

```
typedef struct _PRDINFO {  
    CHAR        szName[PDLEN+1];    /* Name of logical address */  
    CHAR        szUserName[UNLEN+1]; /* User who submitted job */  
    USHORT      uJobId;              /* Identity of current job */  
    USHORT      fsStatus;            /* Print destination status */  
}
```

```

        PSZ        pszStatus;           /* Print destination comment while
                                         printing */
        USHORT     time;                /* Time job has been printing
                                         (minutes) */
    } PRDINFO;

```

PRDINFO3

Print-destination information structure (level 3).

```

typedef struct _PRDINFO3 {
    PSZ        pszPrinterName;  /* Printer name */
    PSZ        pszUserName;     /* User who submitted job */
    PSZ        pszLogAddr;      /* Logical address */
    USHORT     uJobId;          /* Identity of current job */
    USHORT     fsStatus;        /* Print destination status */
    PSZ        pszStatus;       /* Print destination comment while printing
                                */
    PSZ        pszComment;      /* Printer description */
    PSZ        pszDrivers;      /* Drivers supported by printer */
    USHORT     time;            /* Time job has been printing (minutes) */
    USHORT     usTimeOut;       /* Device timeout (seconds) */
} PRDINFO3;

```

PRIDINFO

Print-job identification-number information structure.

```

typedef struct _PRIDINFO {
    USHORT     uJobId;          /* Print-job identification number */
    CHAR        szServer[CNLEN+1]; /* Server */
    CHAR        szQName[QLEN+1];  /* Queue name */
    CHAR        pad_1;           /* Pad character */
} PRIDINFO;

```

PRJINFO

Print-job information structure (level 1).

```

typedef struct _PRJINFO {
    USHORT     uJobId;          /* Job identification number */
    CHAR        szUsername[UNLEN+1]; /* User who submitted the job */
    CHAR        pad_1;          /* Pad character */
    CHAR        szNotifyName[CNLEN+1]; /* Messaging alias for print alert
                                */
    CHAR        szDataType[DTLEN+1]; /* Data type of submitted file */
    PSZ        pszParms;         /* Parameters */
    USHORT     uPosition;        /* Job position in queue */
    USHORT     fsStatus;        /* Job status */
    PSZ        pszStatus;       /* Status comment */
    ULONG      ulSubmitted;      /* Time job submitted */
} PRJINFO;

```

```

        ULONG        uSize;                /* Print-job size (bytes) */
        PSZ          pszComment;           /* Comment string about print job
                                           */
    } PRJINFO;

```

PRJINFO2

Print-job information structure.

This structure provides a subset of the information supplied by *PRJINFO3*. It minimizes the storage required for job-information retrieval, and is sufficient for most uses.

```

typedef struct _PRJINFO2 {
    USHORT        uJobId;                /* Job identification number */
    USHORT        uPriority;             /* Job priority */
    PSZ          pszUserName;           /* User who submitted the job */
    USHORT        uPosition;            /* Job position in queue */
    USHORT        fsStatus;             /* Job status */
    ULONG        uSubmitted;            /* Time job submitted */
    ULONG        uSize;                /* Print-job size (bytes) */
    PSZ          pszComment;           /* Comment string */
    PSZ          pszDocument;          /* Document name */
} PRJINFO2;

```

PRJINFO3

Print-job information structure.

This structure is used when complete job details are required. A subset of this information is supplied by *PRJINFO2*.

```

typedef struct _PRJINFO3 {
    USHORT        uJobId;                /* Job identification number */
    USHORT        uPriority;             /* Job priority */
    PSZ          pszUserName;           /* User who submitted the job */
    USHORT        uPosition;            /* Job position in queue */
    USHORT        fsStatus;             /* Job status */
    ULONG        uSubmitted;            /* Time job submitted */
    ULONG        uSize;                /* Print-job size (bytes) */
    PSZ          pszComment;           /* Comment string */
    PSZ          pszDocument;          /* Document name */
    PSZ          pszNotifyName;        /* Messaging alias for print alert */
    PSZ          pszDataType;          /* Data type of submitted file */
    PSZ          pszParms;             /* Parameters */
    PSZ          pszStatus;            /* Status comment */
    PSZ          pszQueue;             /* Queue name */
    PSZ          pszQProcName;         /* Queue processor */
    PSZ          pszQProcParms;        /* Queue-processor parameters */
    PSZ          pszDriverName;        /* Driver name */
    PDRIVDATA     pDriverData;         /* Driver data */
    PSZ          pszPrinterName;       /* Printer name */
} PRJINFO3;

```

```
} PRJINFO3;
```

PRQINFO

Print-queue information structure for existing applications (level 1).

```
typedef struct _PRQINFO {  
    CHAR        szName[QNLLEN+1];    /* Queue name */  
    CHAR        pad_1;                /* Pad character */  
    USHORT      uPriority;             /* Queue priority */  
    USHORT      uStartTime;           /* Minutes after midnight when queue  
                                      becomes active */  
    USHORT      uUntilTime;           /* Minutes after midnight when queue  
                                      ceases to be active */  
    PSZ         pszSepFile;           /* Separator-page file */  
    PSZ         pszPrProc;            /* Default queue-processor for queue */  
    PSZ         pszDestinations;      /* Print destinations connected to queue  
                                      */  
    PSZ         pszParms;             /* Queue parameters */  
    PSZ         pszComment;           /* Queue description */  
    USHORT      fsStatus;             /* Queue status */  
    USHORT      cJobs;               /* Number of jobs in queue */  
}  
PRQINFO;
```

PRQINFO3

Print-queue information structure.

This structure is used at information levels 3 and 4.

```
typedef struct _PRQINFO3 {  
    PSZ         pszName;              /* Queue name */  
    USHORT      uPriority;             /* Queue priority */  
    USHORT      uStartTime;           /* Minutes after midnight when queue  
                                      becomes active */  
    USHORT      uUntilTime;           /* Minutes after midnight when queue  
                                      ceases to be active */  
    USHORT      pad1;                /* Pad character */  
    PSZ         pszSepFile;           /* Separator-page file */  
    PSZ         pszPrProc;            /* Default queue-processor */  
    PSZ         pszParms;             /* Queue parameters */  
    PSZ         pszComment;           /* Queue description */  
    USHORT      fsStatus;             /* Queue status */  
    USHORT      cJobs;               /* Number of jobs in queue */  
    PSZ         pszPrinters;          /* Print destinations connected to queue  
                                      */  
    PSZ         pszDriverName;        /* Default device driver */  
    PDRIVDATA   pDriverData;          /* Default driver data */  
}  
PRQINFO3;
```

RENDERFILE

File-rendering structure.

```
typedef struct _RENDERFILE {  
    HWND      hwndDragFiles;    /* Conversation handle */  
    HSTR      hstrSource;       /* Handle to source file name */  
    HSTR      hstrTarget;       /* Handle to target file name */  
    BOOL      fMove;            /* Operation */  
    USHORT    usReserved;       /* Reserved */  
} RENDERFILE;
```

SPLERR

Error value in the range 0 to 65 535.

```
typedef USHORT SPLERR;
```

Chapter 3.1. Drag Function Calls

Summary of Changes

New	Updated	Section Title
✓		DrgAcceptDroppedFiles
✓		DrgAccessDraginfo
✓		DrgAddStrHandle
✓		DrgAllocDraginfo
✓		DrgAllocDragtransfer
✓		DrgDeleteDraginfoStrHandles
✓		DrgDeleteStrHandle
✓		DrgDrag
✓		DrgDragFiles
✓		DrgFreeDraginfo
✓		DrgFreeDragtransfer
✓		DrgGetPS
✓		DrgPostTransferMsg
✓		DrgPushDraginfo
✓		DrgQueryDragitem
✓		DrgQueryDragitemCount
✓		DrgQueryDragitemPtr
✓		DrgQueryNativeRMF
✓		DrgQueryNativeRMFLen
✓		DrgQueryStrName
✓		DrgQueryStrNameLen
✓		DrgQueryTrueType
✓		DrgQueryTrueTypeLen
✓		DrgReleasePS
✓		DrgSendTransferMsg
✓		DrgSetDragImage
✓		DrgSetDragitem
✓		DrgSetDragPointer
✓		DrgVerifyNativeRMF
✓		DrgVerifyRMF
✓		DrgVerifyTrueType
✓		DrgVerifyType
✓		DrgVerifyTypeSet

Insert this new chapter after **Chapter 3**.

DrgAcceptDroppedFiles

```
#define INCL_WINSTDDRAG

BOOL    fSuccess = DrgAcceptDroppedFiles (Hwnd, pszPath, pszTypes,
                                           usDefaultOp, usReserved)

HWND    Hwnd;        /* Handle of calling window */
PSZ     pszPath;     /* Directory in which to place the dropped files */
PSZ     pszTypes;    /* A list of types that are acceptable to the drop */
USHORT  usDefaultOp; /* Default drag operation for this window */
USHORT  usReserved;  /* Reserved */

BOOL    fSuccess;    /* Success indicator */
```

DrgAccessDraginfo

```
#define INCL_WINSTDDRAG

BOOL    fSuccess = DrgAccessDraginfo (pDraginfo)

PDRAGINFO pDraginfo; /* Drag-information structure */

BOOL    fSuccess;    /* Success indicator */
```

DrgAddStrHandle

```
#define INCL_WINSTDDRAG

HSTR    Hstr = DrgAddStrHandle (pszString)

PSZ     pszString; /* String for which a handle is to be created */

HSTR    Hstr;      /* Handle of the string */
```

DrgAllocDraginfo

```
#define INCL_WINSTDDRAG

PDRAGINFO pDraginfo = DrgAllocDraginfo (cDitem)

USHORT    cDitem; /* Number of objects being dragged */

PDRAGINFO pDraginfo; /* DRAGINFO structure */
```

DrgAllocDragtransfer

```
#define INCL_WINSTDDRAG

PDRAGTRANSFER pResult = DrgAllocDragtransfer (cdxfer)

USHORT    cdxfer; /* The number of DRAGTRANSFER structures to be
                  allocated */

PDRAGTRANSFER pResult; /* Pointer to an array of DRAGTRANSFER structures */
```

DrgDeleteDraginfoStrHandles

```
#define INCL_WINSTDDRAG

BOOL      fSuccess = DrgDeleteDraginfoStrHandles (pDraginfo)

PDRAGINFO pDraginfo; /* The DRAGINFO structure containing string handles to
                        delete */

BOOL      fSuccess; /* Success indicator */
```

DrgDeleteStrHandle

```
#define INCL_WINSTDDRAG

BOOL fSuccess = DrgDeleteStrHandle (Hstr)

HSTR Hstr; /* The string handle to delete */

BOOL fSuccess; /* Success indicator */
```

DrgDrag

```
#define INCL_WINSTDDRAG

HWND      hwndDest = DrgDrag (hwndSource, pDraginfo, pdimg, cding,
                               vkTerminate, pReserved)

HWND      hwndSource; /* Handle of the window calling DrgDrag */
PDRAGINFO pDraginfo; /* A DRAGINFO structure */
PDRAGIMAGE pdimg; /* Pointer to an array of DRAGIMAGE structures */
USHORT    cding; /* Size of the pdimg array */
SHORT     vkTerminate; /* Mouse button that terminates the drag operation */
PVOID     pReserved; /* Reserved */

HWND      hwndDest; /* Handle of window on which the dragged objects were
                    dropped */
```

DrgDragFiles

```
#define INCL_WINSTDDRAG

BOOL      fSuccess = DrgDragFiles (Hwnd, pFiles, pTypes, pTargets, cFiles,
                                   hptrDrag, vkTerminate, fSourceRender,
                                   ulReserved)

HWND      Hwnd; /* Handle of calling window */
PAPSZ     pFiles; /* The names of the files to be dragged */
PAPSZ     pTypes; /* The file types of the files to be dragged */
PAPSZ     pTargets; /* The target file names */
USHORT    cFiles; /* Number of files to be dragged */
HPOINTER  hptrDrag; /* Icon to display during the drag */
USHORT    vkTerminate; /* Button that terminates the drag */
BOOL      fSourceRender; /* Flag to indicate whether the source must perform
                          the move or copy */

ULONG     ulReserved; /* Reserved */

BOOL      fSuccess; /* Success indicator */
```

DrgFreeDraginfo

```
#define INCL_WINSTDDRAG
BOOL      fSuccess = DrgFreeDraginfo (pDraginfo)
PDRAGINFO pDraginfo; /* A DRAGINFO structure */
BOOL      fSuccess; /* Success indicator */
```

DrgFreeDragtransfer

```
#define INCL_WINSTDDRAG
BOOL      Rc = DrgFreeDragtransfer (pdxfer)
PDRAGTRANSFER pdxfer; /* The DRAGTRANSFER structures to be freed */
BOOL      Rc; /* Return code */
```

DrgGetPS

```
#define INCL_WINSTDDRAG
HPS Hps = DrgGetPS (Hwnd)
HWND Hwnd; /* Handle of the window for which presentation space is required
            */
HPS Hps; /* Presentation-space handle used for drawing in the window */
```

DrgPostTransferMsg

```
#define INCL_WINSTDDRAG
BOOL      fSuccess = DrgPostTransferMsg (hwndTo, usMsgid, pdxfer, usFs,
                                         usReserved, fRetry)
HWND      hwndTo; /* Window handle to which the message is to be
                  posted */
USHORT     usMsgid; /* Identifier of the message to be posted */
PDRAGTRANSFER pdxfer; /* The DRAGTRANSFER structure */
USHORT     usFs; /* The flags to be passed in the param2 parameter
                 of the message */
USHORT     usReserved; /* Reserved */
BOOL      fRetry; /* Retry indicator */
BOOL      fSuccess; /* Success indicator */
```

DrgPushDraginfo

```
#define INCL_WINSTDDRAG
BOOL      fSuccess = DrgPushDraginfo (pDraginfo, hwndDest)
PDRAGINFO pDraginfo; /* DRAGINFO structure */
HWND      hwndDest; /* Handle of the window whose process is to be given
                    access to a DRAGINFO structure */
BOOL      fSuccess; /* Success indicator */
```

DrgQueryDragitem

```
#define INCL_WINSTDDRAG

BOOL      fSuccess = DrgQueryDragitem (pDraginfo, cbBuffer, pDragitem, iItem)
PDRAGINFO pDraginfo; /* The DRAGINFO structure from which the DRAGITEM
                      structure is obtained */
USHORT    cbBuffer; /* Maximum number of bytes to copy */
PDRAGITEM pDragitem; /* Buffer into which the DRAGITEM structure is copied */
USHORT    iItem; /* Zero-based index of the DRAGITEM to be returned */
BOOL      fSuccess; /* Success indicator */
```

DrgQueryDragitemCount

```
#define INCL_WINSTDDRAG

USHORT    cDitem = DrgQueryDragitemCount (pDraginfo)
PDRAGINFO pDraginfo; /* The DRAGINFO structure for which the number of
                      dragged objects is requested */
USHORT    cDitem; /* Number of objects being dragged */
```

DrgQueryDragitemPtr

```
#define INCL_WINSTDDRAG

PDRAGITEM pDragitem = DrgQueryDragitemPtr (pDraginfo, usn)
PDRAGINFO pDraginfo; /* The DRAGINFO structure from which the DRAGITEM
                      structure is obtained */
USHORT    usn; /* Zero-based index of the DRAGITEM structure for which
               the pointer is to be returned */
PDRAGITEM pDragitem; /* The DRAGITEM structure */
```

DrgQueryNativeRMF

```
#define INCL_WINSTDDRAG

BOOL      fSuccess = DrgQueryNativeRMF (pDragitem, cbBuflen, pchBuffer)
PDRAGITEM pDragitem; /* DRAGITEM structure whose native rendering mechanism
                      and format is to be obtained */
USHORT    cbBuflen; /* Maximum number of bytes to copy to the buffer */
PCHAR     pchBuffer; /* Buffer in which the null-terminated string is to be
                      returned */
BOOL      fSuccess; /* Success indicator */
```

DrgQueryNativeRMFLen

```
#define INCL_WINSTDDRAG
USHORT      usLength = DrgQueryNativeRMFLen (pDragitem)
PDRAGITEM   pDragitem; /* DRAGITEM structure whose native rendering mechanism
                        and format string length are to be obtained */
USHORT      usLength; /* String length of the ordered pair */
```

DrgQueryStrName

```
#define INCL_WINSTDDRAG
USHORT      usLength = DrgQueryStrName (Hstr, cbBuflen, pszBuffer)
HSTR        Hstr;      /* Handle to a string */
USHORT      cbBuflen; /* Maximum number of bytes to copy */
PSZ         pszBuffer; /* Buffer where the null-terminated string is returned */
USHORT      usLength; /* Number of bytes written to Buffer */
```

DrgQueryStrNameLen

```
#define INCL_WINSTDDRAG
USHORT      cLength = DrgQueryStrNameLen (Hstr)
HSTR        Hstr;      /* String handle */
USHORT      cLength; /* String length */
```

DrgQueryTrueType

```
#define INCL_WINSTDDRAG
BOOL        fSuccess = DrgQueryTrueType (pDragitem, cbBuflen, pszBuffer)
PDRAGITEM   pDragitem; /* DRAGITEM structure whose true type is to be obtained
                        */
USHORT      cbBuflen; /* Maximum number of bytes to copy to the buffer */
PSZ         pszBuffer; /* Buffer in which the null-terminated string is to be
                        returned */
BOOL        fSuccess; /* Success indicator */
```

DrgQueryTrueTypeLen

```
#define INCL_WINSTDDRAG
USHORT      usLength = DrgQueryTrueTypeLen (pDragitem)
PDRAGITEM   pDragitem; /* DRAGITEM structure whose true type length is to be
                        obtained */
USHORT      usLength; /* String length */
```

DrgReleasePS

```
#define INCL_WINSTDDRAG

BOOL fSuccess = DrgReleasePS (Hps)

HPS Hps; /* Handle of the presentation space to release */

BOOL fSuccess; /* Success indicator */
```

DrgSendTransferMsg

```
#define INCL_WINSTDDRAG

MRESULT mresReply = DrgSendTransferMsg (hwndTo, usMsgid, mpParam1, mpParam2)

HWND hwndTo; /* Window handle to which the message is to be sent */
USHORT usMsgid; /* Identifier of the message to be sent */
MPARAM mpParam1; /* mp1 for the message */
MPARAM mpParam2; /* mp2 for the message */

MRESULT mresReply; /* Message-return data */
```

DrgSetDragImage

```
#define INCL_WINSTDDRAG

BOOL fSuccess = DrgSetDragImage (pDraginfo, pdimg, cding, pReserved)

PDRAGINFO pDraginfo; /* DRAGINFO structure representing the drag operation
for which the mouse pointer is to be set */
PDRAGIMAGE pdimg; /* Pointer to an array of DRAGIMAGE structures */
USHORT cding; /* Size of the pdimg array */
PVOID pReserved; /* Reserved */

BOOL fSuccess; /* Success indicator */
```

DrgSetDragitem

```
#define INCL_WINSTDDRAG

BOOL fSuccess = DrgSetDragitem (pDraginfo, pDragitem, cbBuffer, iItem)

PDRAGINFO pDraginfo; /* DRAGINFO structure in which to place the DRAGITEM */
PDRAGITEM pDragitem; /* DRAGITEM structure to place in DRAGINFO */
USHORT cbBuffer; /* Size of the DRAGITEM addressed by Dragitem */
USHORT iItem; /* Zero-based index of the DRAGITEM to be set */

BOOL fSuccess; /* Success indicator */
```

DrgSetDragPointer

```

#define INCL_WINSTDDRAG
BOOL      fResult = DrgSetDragPointer (pDraginfo, hptrHandle)
PDRAGINFO pDraginfo; /* Pointer to the DRAGINFO structure to be used for
                        this drag */
HPOINTER  hptrHandle; /* Handle to the pointer to use */
BOOL      fResult;    /* Success indicator */

```

DrgVerifyNativeRMF

```

#define INCL_WINSTDDRAG
BOOL      fValid = DrgVerifyNativeRMF (pDragitem, pszRMF)
PDRAGITEM pDragitem; /* DRAGITEM structure whose native rendering mechanism
                        and format are to be verified */
PSZ       pszRMF;     /* A string specifying the rendering mechanism and
                        format */
BOOL      fValid;     /* Validity indicator */

```

DrgVerifyRMF

```

#define INCL_WINSTDDRAG
BOOL      fValid = DrgVerifyRMF (pDragitem, pszMech, pszFormat)
PDRAGITEM pDragitem; /* DRAGITEM structure whose native rendering mechanism
                        and format is to be validated */
PSZ       pszMech;    /* A string specifying the rendering mechanism to search
                        for */
PSZ       pszFormat;  /* A string specifying the rendering format to search
                        for */
BOOL      fValid;     /* Validity indicator */

```

DrgVerifyTrueType

```

#define INCL_WINSTDDRAG
BOOL      fValid = DrgVerifyTrueType (pDragitem, pszType)
PDRAGITEM pDragitem; /* DRAGITEM structure whose true type is to be verified
                        */
PSZ       pszType;    /* A string specifying a type */
BOOL      fValid;     /* Validity indicator */

```

DrgVerifyType

```

#define INCL_WINSTDDRAG
BOOL      fValid = DrgVerifyType (pDragitem, pszType)
PDRAGITEM pDragitem; /* DRAGITEM structure whose type is to be verified */
PSZ       pszType;    /* A string specifying the types to search for */
BOOL      fValid;     /* Success indicator */

```

DrgVerifyTypeSet

```
#define INCL_WINSTDDRAG

BOOL      fMatch = DrgVerifyTypeSet (pDragitem, pszType, cbBuflen, pszBuffer)

PDRAGITEM pDragitem; /* DRAGITEM structure whose type is to be verified */
PSZ       pszType;   /* A string specifying the types to search for */
USHORT    cbBuflen;  /* Size of the return buffer */
PSZ       pszBuffer; /* Buffer where the intersection string is returned */
BOOL      fMatch;    /* Match indicator */
```

Chapter 7. Spooler Function Calls

Summary of Changes

New	Updated	Section Title
✓		DosPrintDestAdd
✓		DosPrintDestControl
✓		DosPrintDestDel
✓		DosPrintDestEnum
✓		DosPrintDestGetInfo
✓		DosPrintDestSetInfo
✓		DosPrintJobContinue
✓		DosPrintJobDel
✓		DosPrintJobEnum
✓		DosPrintJobGetId
✓		DosPrintJobGetInfo
✓		DosPrintJobPause
✓		DosPrintJobSetInfo
✓		DosPrintQAdd
✓		DosPrintQContinue
✓		DosPrintQDel
✓		DosPrintQEnum
✓		DosPrintQGetInfo
✓		DosPrintQPause
✓		DosPrintQPurge
✓		DosPrintQSetInfo

On page 7-1, at the top of the page, insert the following:

DosPrintDestAdd

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR rc = DosPrintDestAdd (pszComputerName, usLevel, pbBuf, cbBuf)
PSZ     pszComputerName; /* Name of computer where destination is to be added
                          */
USHORT  usLevel;         /* Level of detail provided */
PBYTE   pbBuf;           /* Data structure */
USHORT  cbBuf;           /* Size, in bytes, of data structure */
SPLERR  rc;
```

DosPrintDestControl

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR rc = DosPrintDestControl (pszComputerName, pszPrinterName, usCONTROL)
PSZ    pszComputerName; /* Name of computer where destination is to be
                        controlled */
PSZ    pszPrinterName;  /* Name of Print Destination */
USHORT usCONTROL;       /* Operation to perform */
SPLERR rc;
```

DosPrintDestDel

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR rc = DosPrintDestDel (pszComputerName, pszPrinterName)
PSZ    pszComputerName; /* Name of computer where destination is to be
                        deleted */
PSZ    pszPrinterName;  /* Name of Print Destination */
SPLERR rc;
```

DosPrintDestEnum

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR rc = DosPrintDestEnum (pszComputerName, usLevel, pbBuf, cbBuf,
                             pcReturned, pcTotal)
PSZ    pszComputerName; /* Name of computer where destinations are to be
                        listed */
USHORT usLevel;         /* Level of detail required */
PBYTE  pbBuf;           /* Data structure */
USHORT cbBuf;           /* Size, in bytes, of data structure */
PUSHORT pcReturned;     /* Number of entries returned */
PUSHORT pcTotal;        /* Number of entries available */
SPLERR rc;
```

DosPrintDestGetInfo

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR rc = DosPrintDestGetInfo (pszComputerName, pszPrinterName, usLevel,
                                 pbBuf, cbBuf, pcbNeeded)
PSZ    pszComputerName; /* Name of computer where destination is to be
                        queried */
PSZ    pszPrinterName;  /* Name of Print Destination */
USHORT usLevel;         /* Level of detail required */
PBYTE  pbBuf;           /* Data structure */
USHORT cbBuf;           /* Size, in bytes, of data structure */
PUSHORT pcbNeeded;      /* Number of bytes of information available */
SPLERR rc;
```

DosPrintDestSetInfo

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintDestSetInfo (pszComputerName, pszPrinterName, usLevel,
                                pbBuf, cbBuf, usParmNum)

PSZ    pszComputerName; /* Name of computer where destination is to be
                        modified */
PSZ    pszPrinterName;  /* Name of Print Destination */
USHORT usLevel;         /* Level of detail */
PBYTE  pbBuf;           /* Data structure */
USHORT  cbBuf;          /* Size, in bytes, of data structure */
USHORT  usParmNum;      /* Parameter number */

SPLERR rc;
```

DosPrintJobContinue

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintJobContinue (pszComputerName, idJob)

PSZ    pszComputerName; /* Name of computer where job is to be continued */
USHORT idJob;           /* Job identification number */

SPLERR rc;
```

DosPrintJobDel

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintJobDel (pszComputerName, idJob)

PSZ    pszComputerName; /* Name of computer where job is to be deleted */
USHORT idJob;           /* Job identification number */

SPLERR rc;
```

DosPrintJobEnum

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintJobEnum (pszComputerName, pszQueueName, usLevel, pbBuf,
                             cbBuf, pcReturned, pcTotal)

PSZ    pszComputerName; /* Name of computer where jobs are to be listed */
PSZ    pszQueueName;    /* Queue name */
USHORT usLevel;         /* Level of detail required */
PBYTE  pbBuf;           /* Data structure */
USHORT  cbBuf;          /* Size, in bytes, of data structure */
PUSHORT pcReturned;     /* Number of entries returned */
PUSHORT pcTotal;        /* Number of entries available */

SPLERR rc;
```

DosPrintJobGetId

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR    rc = DosPrintJobGetId (hFile, pInfo, cbInfo)
HFILE     hFile; /* Handle of redirected print device */
PPRIDINFO pInfo; /* Data structure */
USHORT    cbInfo; /* Size, in bytes, of data structure */
SPLERR    rc;
```

DosPrintJobGetInfo

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR    rc = DosPrintJobGetInfo (pszComputerName, idJob, usLevel, pbBuf,
                                   cbBuf, pcbNeeded)
PSZ       pszComputerName; /* Name of computer where print job is to be queried
                             */
USHORT    idJob;           /* Job identification number */
USHORT    usLevel;        /* Level of detail required */
PBYTE     pbBuf;          /* Data structure */
USHORT    cbBuf;          /* Size, in bytes, of data structure */
PUSHORT    pcbNeeded;      /* Size in bytes of available information */
SPLERR    rc;
```

DosPrintJobPause

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR    rc = DosPrintJobPause (pszComputerName, idJob)
PSZ       pszComputerName; /* Name of computer where job is to be paused */
USHORT    idJob;           /* Job identification number */
SPLERR    rc;
```

DosPrintJobSetInfo

```
#define INCL_SPL    /* Or use INCL_PM */
SPLERR    rc = DosPrintJobSetInfo (pszComputerName, idJob, usLevel, pbBuf,
                                   cbBuf, usParmNum)
PSZ       pszComputerName; /* Name of computer where job is to be modified */
USHORT    idJob;           /* Job identification number */
USHORT    usLevel;        /* Level of detail required */
PBYTE     pbBuf;          /* Data structure */
USHORT    cbBuf;          /* Size, in bytes, of data structure */
USHORT    usParmNum;      /* Parameter number */
SPLERR    rc;
```

DosPrintQAdd

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQAdd (pszComputerName, usLevel, pbBuf, cbBuf)

PSZ    pszComputerName; /* Name of computer where queue is to be created */
USHORT usLevel;          /* Level of detail provided */
PBYTE  pbBuf;            /* Data structure */
USHORT cbBuf;            /* Size, in bytes, of data structure */

SPLERR rc;
```

DosPrintQContinue

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQContinue (pszComputerName, pszQueueName)

PSZ    pszComputerName; /* Name of computer where queue is to be continued */
PSZ    pszQueueName;    /* Queue name */

SPLERR rc;
```

DosPrintQDel

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQDel (pszComputerName, pszQueueName)

PSZ    pszComputerName; /* Name of computer where queue is to be deleted */
PSZ    pszQueueName;    /* Queue name */

SPLERR rc;
```

DosPrintQEnum

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQEnum (pszComputerName, usLevel, pbBuf, cbBuf,
                          pcReturned, pcTotal)

PSZ    pszComputerName; /* Name of computer where queues are to be listed */
USHORT usLevel;          /* Level of detail */
PBYTE  pbBuf;            /* Data structure */
USHORT cbBuf;            /* Size, in bytes, of data structure */
PUSHORT pcReturned;      /* Number of entries returned */
PUSHORT pcTotal;         /* Total number of entries available */

SPLERR rc;
```

DosPrintQGetInfo

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQGetInfo (pszComputerName, pszQueueName, usLevel, pbBuf,
                              cbBuf, pcbNeeded)

PSZ    pszComputerName; /* Name of computer where queue is to be queried */
PSZ    pszQueueName;    /* Queue name */
USHORT usLevel;         /* Level of detail required */
PBYTE  pbBuf;           /* Data structure */
USHORT  cbBuf;          /* Size, in bytes, of data structure */
PUSHORT pcbNeeded;      /* Size, in bytes, of available information */

SPLERR rc;
```

DosPrintQPause

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQPause (pszComputerName, pszQueueName)

PSZ    pszComputerName; /* Name of computer where queue is to be paused */
PSZ    pszQueueName;    /* Queue name */

SPLERR rc;
```

DosPrintQPurge

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQPurge (pszComputerName, pszQueueName)

PSZ    pszComputerName; /* Name of computer where queue is to be purged */
PSZ    pszQueueName;    /* Queue name */

SPLERR rc;
```

DosPrintQSetInfo

```
#define INCL_SPL    /* Or use INCL_PM */

SPLERR rc = DosPrintQSetInfo (pszComputerName, pszQueueName, usLevel, pbBuf,
                              cbBuf, usParmNum)

PSZ    pszComputerName; /* Name of computer where queue is to be modified */
PSZ    pszQueueName;    /* Queue name */
USHORT usLevel;         /* Level of detail required */
PBYTE  pbBuf;           /* Data structure */
USHORT  cbBuf;          /* Size, in bytes, of data structure */
USHORT  usParmNum;      /* Parameter number */

SPLERR rc;
```

Chapter 10. Functions Supplied by Applications

Summary of Changes

New	Updated	Section Title
	✓	JournalPlaybackHook

JournalPlaybackHook

On page 10-1, replace the language binding with the following:

```
LONG  lTime = JournalPlaybackHook (hab, fSkip, pqmsg)
HAB    hab; /* Anchor-block handle */
BOOL   fSkip; /* Indicator as to whether the next message should be played
              back */
PQMSG  pqmsg; /* Data structure where the message to be played back is
              returned */
LONG   lTime; /* Waiting time */
```

Presentation Manager COBOL/2 Bindings Reference

Chapter 1. Introduction

Summary of Changes

New	Updated	Section Title
✓		Named Constants

Named Constants

On page 1-2, replace the Miscellaneous Definitions section with the following:

Various constants needed by COBOL Presentation Manager applications are provided as named constants in the COBOL header files, PM.CIN and OS2.CIN. Therefore, these must be included at compile time, by means of the COPY statement for example.

In particular, the following definitions are provided to avoid a clash with COBOL reserved words:

```
77 FALSE-VALUE  VALUE IS 0.
```

```
77 TRUE-VALUE   VALUE IS 1.
```

Chapter 2. Data Types

Summary of Changes

New	Updated	Section Title
	✓	COUNT4
	✓	COUNT4B
	✓	LENGTH4
	✓	LONG
	✓	MARGSTRUCT
	✓	OVERFLOW
	✓	PROPERTY4
	✓	ULONG
	✓	WIDTH4

On page 2-1, at the top of the page, add the following:

Note: For the data types of the individual structure elements of those data types that map to *HSTRUCT*, refer to Chapter 2 of the *OS/2 Version 1.2 Presentation Manager Programming Reference: Volume 1*.

COUNT4

On page 2-6, change the first line to the following:

Count in the range 0 through 999 999 999.

COUNT4B

On page 2-6, change the first line to the following:

Count of bytes, in the range of 0 through 999 999 999.

LENGTH4

On page 2-16, change the first line to the following:

Length value, in the range 0 through 999 999 999.

LONG

On page 2-16, change the first line to the following:

Signed integer in the range –999 999 999 through 999 999 999.

MARGSTRUCT

On page 2-17, this data type is now named *MLEMARGSTRUCT*.

OVERFLOW

On page 2-19, this data type is now named *MLEOVERFLOW*.

PROPERTY4

On page 2-21, change the first line to the following:

Property value. Up to 999 999 999 different properties are available.

ULONG

On page 2-27, change the first line to the following:

Unsigned integer in the range 0 through 999 999 999.

WIDTH4

On page 2-28, change the first line to the following:

Width in the range —999 999 999 through 999 999 999.

Chapter 8. Window Function Calls

Summary of Changes

New	Updated	Section Title
	√	WinQuerySwitchList — WMQSLI

WinQuerySwitchList — WMQSLI

On page 8-36, add the following:

Note: The **Count** field of the **SwitchEntries** parameter must be initialized to the number of switch entries for which space is available.

Presentation Manager FORTRAN/2 Bindings Reference

Chapter 1. Introduction

Summary of Changes

New	Updated	Section Title
✓		Named Constants

Named Constants

On page 1-3, replace the Miscellaneous Definitions section with the following:

Various constants needed by FORTRAN Presentation Manager applications are provided as named constants in the FORTRAN header files, PM.FIN and OS2.FIN. Therefore, these must be included at compile time, by means of the INCLUDE statement for example.

The following definitions are provided in the FORTRAN header file, OS2.FIN:

```
INTEGER*4  FALSE
PARAMETER (FALSE=0)
```

```
INTEGER*4  TRUE
PARAMETER (TRUE=1)
```

```
INTEGER*4  NULL
PARAMETER (NULL=0)
```

Chapter 2. Data Types

Summary of Changes

New	Updated	Section Title
	✓	HPROGARRAY
	✓	MARGSTRUCT
	✓	OVERFLOW

On page 2-1, at the beginning of the page, add the following:

Note: For the data types of the individual structure elements of those data types that map to *HSTRUCT*, refer to Chapter 2 of the *OS/2 Version 1.2 Presentation Manager Programming Reference: Volume 1*.

HPROGARRAY

On page 2-12, delete the lines:

```
C*** DECLAREs for structure components (EQUIVALENCed onto
HPROGA_BLOCK)
```

MARGSTRUCT

On page 2-14, this data type is now named *MLEMARGSTRUCT*.

OVERFLOW

On page 2-17, this data type is now named *MLEOVERFLOW*.

Chapter 6. Profile Function Calls

Summary of Changes

New	Updated	Section Title
	✓	PrfQueryProgramHandle — PRQPHD

PrfQueryProgramHandle — PRQPHD

On page 6-4, change the line that reads:

CHARACTER*(HPROGA_LEN) PROGARRAY

to:

INTEGER*(HPROGA_LEN) PROGARRAY

Chapter 8. Window Function Calls

Summary of Changes

New	Updated	Section Title
	√	WinQuerySwitchList — WMQSLI

WinQuerySwitchList — WMQSLI

On page 8-38, add the following:

Note: The **Count** field of the **SwitchEntries** parameter must be initialized to the number of switch entries for which space is available.

Presentation Manager Macro Assembler/2 Bindings Reference

Chapter 2. Data Types

Summary of Changes

New	Updated	Section Title
	✓	MARGSTRUCT
	✓	OVERFLOW

MARGSTRUCT

On page 2-10, this data type is now named *MLEMARGSTRUCT*.

OVERFLOW

On page 2-12, this data type is now named *MLEOVERFLOW*.

Chapter 9. Window Function Calls

Summary of Changes

New	Updated	Section Title
✓		WinQuerySessionTitle
✓		WinSwitchToProgram

WinQuerySessionTitle

On page 9-30, add this call:

```
EXTRN WinQuerySessionTitle:FAR
INCL_WINSWITCHLIST equ 1 ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hab          ;Anchor-block handle
PUSH  WORD   usSession    ;OS/2 session identity of application whose title is requested
PUSH@  ASCIIZ szTitle     ;Task title
PUSH  WORD   usTitlelen   ;Maximum length of data returnable in bytes
CALL  WinQuerySessionTitle

Returns WORD   usRetCode  ;Return code
```

WinSwitchToProgram

On page 9-44, add this call:

```
EXTRN WinSwitchToProgram:FAR
INCL_WINSWITCHLIST equ 1      ;Or use INCL_WIN or INCL_PM

PUSH  DWORD  hswitchSwHandle ;Switch list entry handle of program to be activated
CALL  WinSwitchToProgram

Returns WORD   usRetCode      ;Return code
```

Chapter 10. Functions Supplied by Applications

Summary of Changes

New	Updated	Section Title
	✓	JournalPlaybackHook

JournalPlaybackHook

On page 10-2, replace the language binding with the following:

```
EXTRN JournalPlaybackHook:FAR
```

```
PUSH    DWORD    hab    ;Anchor-block handle
```

```
PUSH    WORD     fSkip  ;Indicator as to whether the next message should be played back
```

```
PUSH@   OTHER    qmsg   ;Data structure where the message to be played back is returned
```

```
CALL    JournalPlaybackHook
```

Returns DWORD lTime ;Waiting time

Index

C

COBOL data types
COUNT4 33
COUNT4B 33
LENGTH4 33
LONG 33
MARGSTRUCT 34
OVERFLOW 34
PROPERTY4 34
ULONG 34
WIDTH4 34

D

DosPrintDestAdd
C language binding 21
DosPrintDestControl
C language binding 22
DosPrintDestDel
C language binding 22
DosPrintDestEnum
C language binding 22
DosPrintDestGetInfo
C language binding 22
DosPrintDestSetInfo
C language binding 23
DosPrintJobContinue
C language binding 23
DosPrintJobDel
C language binding 23
DosPrintJobEnum
C language binding 23
DosPrintJobGetId
C language binding 24
DosPrintJobGetInfo
C language binding 24
DosPrintJobPause
C language binding 24
DosPrintJobSetInfo
C language binding 24
DosPrintQAdd
C language binding 25
DosPrintQContinue
C language binding 25
DosPrintQDel
C language binding 25
DosPrintQEnum
C language binding 25
DosPrintQGetInfo
C language binding 26
DosPrintQPause
C language binding 26

DosPrintQPurge
C language binding 26
DosPrintQSetInfo
C language binding 26
DRAGIMAGE
C Language binding 3
DRAGINFO
C language binding 4
DRAGITEM
C language binding 4
DRAGTRANSFER
C language binding 5
DrgAcceptDroppedFiles
C language binding 12
DrgAccessDraginfo
C language binding 12
DrgAddStrHandle
C language binding 12
DrgAllocDraginfo
C language binding 12
DrgAllocDragtransfer
C language binding 12
DrgDeleteDraginfoStrHandles
C language binding 13
DrgDeleteStrHandle
C language binding 13
DrgDrag
C language binding 13
DrgDragFiles
C language binding 13
DrgFreeDraginfo
C language binding 14
DrgFreeDragtransfer
C language binding 14
DrgGetPS
C language binding 14
DrgPostTransferMsg
C language binding 14
DrgPushDraginfo
C language binding 14
DrgQueryDragitem
C language binding 15
DrgQueryDragitemCount
C language binding 15
DrgQueryDragitemPtr
C language binding 15
DrgQueryNativeRMF
C language binding 15
DrgQueryNativeRMFLen
C language binding 16
DrgQueryStrName
C language binding 16
DrgQueryStrNameLen
C language binding 16

DrgQueryTrueType
 C language binding 16
DrgQueryTrueTypeLen
 C language binding 16
DrgReleasePS
 C language binding 17
DrgSendTransferMsg
 C language binding 17
DrgSetDragImage
 C language binding 17
DrgSetDragitem
 C language binding 17
DrgSetDragPointer
 C language binding 18
DrgVerifyNativeRMF
 C language binding 18
DrgVerifyRMF
 C language binding 18
DrgVerifyTrueType 18
 C language binding
DrgVerifyType
 C language binding 18
DrgVerifyTypeSet
 C language binding 19

E

ESCSETMODE
 C language binding 5

F

FORTTRAN binding 41

H

HFILE
 C language binding 5
HLIB
 C language binding 5
HPRGARRAY data type
 FORTTRAN binding 41
HSTR
 C language binding 5

J

JournalPlaybackHook 27, 53
 constants, named 31
 FALSE-VALUE 31
 OS2.CIN 31
 PM.CIN 31
 TRUE-VALUE 31

M

MARGSTRUCT
MARGSTRUCT data type
 FORTTRAN binding 41

O

OVERFLOW data type

P

PAPSZ
 C language binding 6
PRDINFO
 C language binding 6
PRDINFO3
 C language binding 7
PrfQueryProgramHandle - PRQPHD 43
PRIDINFO
 C language binding 7
PRJINFO
 C language binding 7
PRJINFO2
 C language binding 8
PRJINFO3
 C language binding 8
PRQINFO
 C language binding 9
PRQINFO3
 C language binding 9

R

RENDERFILE
 C language binding 10

S

SPLERR
 C language binding 10

W

WinQuerySessionTitle 51
WinQuerySwitchList - WMQSLI 35, 45
 constants, named 39
 FALSE (FORTTRAN) 39
 NULL (FORTTRAN) 39
 OS2.FIN 39
 PM.FIN 39
 TRUE (FORTTRAN) 39
WinSwitchToProgram 51

IBM United Kingdom
International Products Limited
PO Box 41, North Harbour
Portsmouth, PO6 3AU
England

Printed in Denmark by
J. H. Schultz Print a/s
Copenhagen

